

Controle de Estoque Utilizando plataforma Web para o Gerenciamento de Tapeçaria.

Selmo Andrei Bobato¹, Rodrigo Souza D'Avila²

¹Tecnologia em Análise e Desenvolvimento de Sistemas – Faculdade Guairacá
CEP 85010-000 – Guarapuava – PR – Brasil

¹selmoandrei@hotmail.com, ²rsdavila@gmail.com

Abstract. *This article aims to present the steps taken to the development of a web system for a tapestry inventory control. The language used to develop the software was the PHP (Hypertext Preprocessor) oriented to objects about the IDE (Integrated Development Environment) NetBeans and the Database Management System MySQL applied to Database MySQL. The system aims to help reducing the problems associated to inventory control and cash flow in a company in the tapestry business.*

Resumo. *Este artigo tem por objetivo apresentar os passos realizados para o desenvolvimento de um sistema web para controle de estoque de uma tapeçaria. A linguagem utilizada para o desenvolvimento do software foi o PHP (Hipertext PreProcessor) orientado a objetos sobre o IDE (Integrated Development Environment) NetBeans e o Sistema de Gerenciamento de Banco de Dados MySQL aplicado com o Banco de dados MySQL. O sistema tem por objetivo auxiliar na redução dos problemas associados com o controle de estoque e fluxo de caixa de uma empresa do ramo de tapeçaria.*

Introdução

Com o passar do tempo à informação adquirida começou a ter um custo para ser gerada, segundo Lotte (2004) para que as atividades administrativas nas empresas sejam desempenhadas de forma eficiente e eficaz, é necessária que seja transmitida de forma mais rápida possível. Portanto a tecnologia da informação está sendo a ferramenta mais utilizada hoje em dia para a competitividade. Nesta competitividade das organizações não esta na quantidade de computadores e sua sofisticação, mas na maneira como a tecnologia esta empregada, assim podendo uma empresa a qual investe menos em equipamentos podendo ter um retorno maior do que empresas que investem fortunas.

O presente estudo tem por finalidade apresentar o ciclo de desenvolvimento e os resultados adquiridos durante o processo de desenvolvimento de um software voltado para uma empresa de prestação de serviços no ramo de tapeçaria com ênfase em fluxo de caixa e controle de estoque.

A empresa cliente do software desenvolvido está no mercado há vários anos, mas em nenhuma circunstancia utilizou sistemas para gerenciassem as suas informações de forma clara e precisa.

Durante o desenvolvimento foram analisados todos os dados principais que a empresa necessitaria obter para possuir uma competitividade maior com as concorrentes. Nesta analise foi possível perceber que o cliente sem o software necessita

de muito tempo para localizar uma informação, a qual durante mais de 20 anos encontra-se apenas em anotações.

Para o desenvolvimento do software serão empregadas as seguintes ferramentas: NetBeans IDE (7.3.1), utilizado como ambiente de programação, o MySQL Workbench, sendo utilizado para a modelagem lógica do banco de dados e o Astah Professional, aplicado para a elaboração dos diagramas propostos pela UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada). A linguagem de programação aplicada no desenvolvimento foi o PHP (*Hypertext Preprocessor*) sobre o paradigma orientado a objetos e o banco de dados MySQL. Bibliotecas gratuitas como *Bootstrap* para criação de layouts, juntamente com a biblioteca *Jquery*.

Fundamentação Teórica

Segundo Rezende (2002) um dos recursos que podem auxiliar no equilíbrio estratégico das organizações é a informação. Juntamente com os avanços proporcionados pela informática surge a Tecnologia da Informação (TI), que tem como papel fundamental proporcionar a redução de custos e agilizar no processamento da informação.

Com o decorrer do tempo a tecnologia obteve um desenvolvimento progressivo e o trabalho que anteriormente era executado de forma manual passou a ser realizado por máquinas capazes de simular o trabalho humano. Um exemplo de invenções tecnológicas que podem ser citadas são a roda e a eletricidade.

Rezende (2002) enfatiza ainda que a TI é um elemento essencial ou um mecanismo sustentador para o desenvolvimento organizacional e pode apresentar os seguintes elementos: eliminar fatores de tempo e de distância; compartilhar recursos; tornar a organização competitiva; obter vantagens competitivas em relação às outras organizações.

As organizações buscam melhorar a excelência operacional no mercado, a fim de buscar maior lucratividade. De acordo com Laudon & Laudon (2007), os sistemas de informações (SI) podem ser considerados uma das ferramentas de suma importância que abrangem altos níveis de eficiência e produtividade nas operações, principalmente quando associadas com mudanças no comportamento da administração e nas práticas do negócio.

Pressman (2006), descreve que aplicações e sistemas baseados na Web podem trazer benefícios para um negócio, afinal, são consideradas ferramentas computacionais aprimoradas que não são capazes apenas de oferecerem funções mínimas para o usuário final, mas são também capazes de ser integradas com bancos de dados corporativos e se tornarem aplicações adequadas para os negócios.

Ainda de acordo com Pressman (2006), as aplicações possuem atributos como: a concentração em redes (pertence a uma rede com Internet que servirá para a comunicação dos usuários com o sistema web); evolução continuada (é diferente do software convencional, essas aplicações evoluem de forma contínua); imediatismo (exibem um prazo de colocação no mercado em questão de dias ou até mesmo semanas); segurança (fornece métodos seguros de transmissão de dados); design (o aspecto da WebApp possui uma estética que valoriza na sua interatividade e visualização);

Para que a construção de páginas ou sistemas web obtenha um padrão de desenvolvimento é necessário se utilizar a linguagem de marcação *Hypertext Markup Language* (HTML), nesse contexto Deitel (2001) define que a linguagem HTML é aplicada para auxiliar na formatação de textos e informações, contudo ainda serve para estruturar os elementos de uma página de uma forma que o navegador web (browser) possa exibi-las.

Para que as páginas de um sistema web se tornem dinâmicas é necessário utilizar uma linguagem de programação. No presente estudo, a linguagem de programação aplicada foi a linguagem *Hypertext PreProcessor* (PHP), sobre o paradigma de programação orientada a objetos.

Essa linguagem ainda pode ser combinada com outras linguagens que permitem possibilitar uma interface dinâmica e atrativa. Entre essas linguagens se tem a linguagem *Cascading Style Sheets* (CSS) que para Jobstraibizer (2009), é uma linguagem de estilo definida como Folhas de Estilos em Cascata ou de formatação que auxilia na construção de layouts das páginas web. Permitindo ainda que seja possível a projeção de sistemas com técnicas diferentes das convencionais propostas. Possibilitando ainda uma redução de tempo durante o desenvolvimento e deixando o layout padrão para todas as páginas. Esta linguagem ainda permite auxiliar na separação entre o formato e o conteúdo de um documento.

De acordo com Niederauer (2004), a linguagem PHP é voltada pra a Internet que possibilita o desenvolvimento de sites dinâmicos. Sites que eram estáticos, feitos de HTML puro podem se tornar sites interativos. Contudo, essa linguagem permite sua implementação de forma gratuita e de código aberto, é embutida com a linguagem HTML, diversos bancos de dados podem ser suportados e ainda possui portabilidade para ser executado em sistemas operacionais como Linux, Unix ou Windows.

De acordo com Welling e Thomson (2005), a linguagem PHP possui algumas vantagens em relação as demais linguagens utilizadas tais como: alto desempenho, interfaces para muitos sistemas diferentes de bancos de dados, bibliotecas integradas para muitas tarefas comuns da Web, baixo custo, facilidade de aprender e utilizar, suporte orientado a objetos, disponibilidade de código fonte e disponibilidade de suporte.

A criação de sistemas ou páginas web através da linguagem PHP permite escolher entre a programação procedural (estruturada) e a programação orientada a objetos. De acordo com Dall'Oglio (2007), a programação estruturada é baseada na modularização do código, ou seja, tem por finalidade dividir o programa desenvolvido em módulos menores, que na sua estrutura utilizam funções e procedimentos básicos de controle.

Conforme Rodriguez e Ferrante (2000), a programação orientada a objetos permite que os dados sejam tratados como objetos ou como um conjunto de objetos, que na sua composição possuem determinadas características que são definidas pelos usuários do sistema desenvolvido. Esse modelo ainda permite a relação de objetos e atributos, que na sua estrutura dispõem de regras associadas a estes atributos. Nesse tipo de programação é possível eliminar a divisão clássica entre a programação da aplicação e a programação da estrutura de dados, sendo que as funções são atributos dos objetos. Apesar disso, este tipo de desenvolvimento admite desenvolver pedaços de programas

que podem ser utilizados na modelagem de qualquer aplicação onde esse tipo de programa se aplica, através disso existe maneiras de reutilizar e modularizar a escrita.

O software foi desenvolvido seguindo os conceitos da orientação a objetos.

A tabela 1 apresenta as diferenças entre os paradigmas de programação estruturada e a programação orientada a objetos de acordo com Schach (2010).

Tabela 1: Diferenças entre paradigmas da programação estruturada e orientação a objetos

Estruturada	Orientação a Objetos
Técnicas clássicas eram adequadas para elaboração de produtos de escala pequena	Utilização de técnicas orientadas a atributos e operações ao mesmo tempo;
Maior tempo para realizar a manutenção do software	Redução de falhas no produto;
Impossibilidade de utilizar técnicas orientadas a operações ou a atributos (dados) ao mesmo tempo	Redução de tempo e custo no desenvolvimento e na manutenção;
Implementação como um conjunto de módulos, tornando-se uma única unidade.	Redução do nível de complexidade do software e maior qualidade;
	Reutilização de objetos;
	Formado por uma serie de unidades menores e independentes.

Fonte: Schach (2010)

Para que o sistema web fosse interpretado pelos navegadores foi empregada a linguagem auxiliar JavaScript, que para Miyagusku (2007) é uma linguagem de programação de scripts que quando interligada no código HTML, pode possibilitar na criação de paginas dinâmicas que fornecem maior interatividade para quem esta acessando a pagina. Ainda de acordo com Miyagusku (2007), o JavaScript é uma linguagem que precisa ser interpretada, ou seja, não é previamente compilada para se gerar um executável, sendo assim seus códigos consistem de linhas executáveis.

Para que o usuário possa fazer interação de uma pagina web no navegador, foi aplicado técnicas do Ajax. No entendimento de Sampaio (2007), Ajax é utilizado para as criações de paginas web dinâmicas que possibilitam o carregamento de conteúdo assíncrono. Propiciando melhorar a experiência do usuário com o site, evitando que o usuário fique parado aguardando a abertura total do conteúdo. Esse conjunto de técnicas ainda permite fazer com que as paginas se modifiquem e se atualizem sem a necessidade de fazer operação *refresh*.

Buscando proporcionar uma maior interação do usuário com as paginas da web foi utilizada uma biblioteca de código fonte aberto chamada *Jquery*. Conforme Silva (2010), esta biblioteca de JavaScript tem a finalidade de acrescentar interação e dinamismo nas paginas web desenvolvidas, beneficiando ao usuário usabilidade, acessibilidade juntamente com efeitos visuais e animações na aplicação proposta.

Analisando como seria organizado hierarquicamente o código fonte do software desenvolvido, foi aplicado um padrão de projeto chamado MVC (*Model, View, Controller*). De acordo com Melo (2007), esse tipo de arquitetura auxilia na manutenção, organização e reaproveitamento do código possibilitando menor tempo na

revisão do software. Ainda para Melo (2007), esse tipo de organização se divide em três camadas:

- *Model* (Modelo): Essas camadas contem em sua estrutura a logica da aplicação e a manipulação dos dados.
- *View* (Visualização): Esta camada tem a finalidade de administrar os aspectos da visualização.
- *Controller* (Controle): Essa camada tem a finalidade de direcionar o fluxo do aplicativo fazendo conexão entre as camadas de Visualização e de Modelo. Toda a logica como atribuições e validações é encontrada dentro dessa camada.

A figura 1 apresenta como é o processo do padrão MVC.

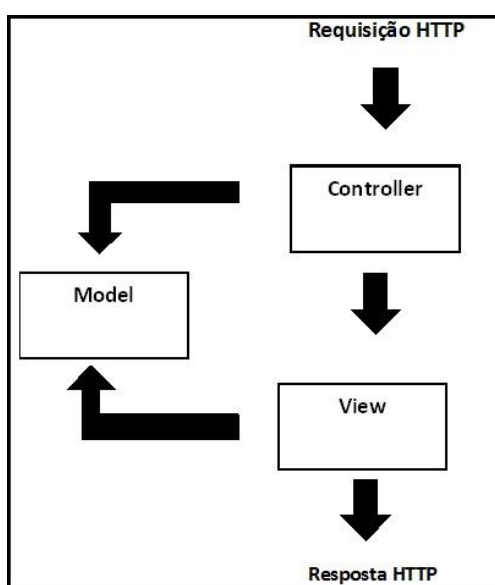


Figura 1 - Conceito de MVC Clássico. Fonte: Melo e Nascimento (2007).

É de suma importância para os sistemas de informações utilizarem uma ferramenta que gere uma quantidade de informações e que ainda esses dados possam ser acessados de forma segura e precisa, sendo assim é necessário empregar um banco de dados para esse tipo de gerenciamento. De acordo com Ramakrishnan e Gehrke (2008), um sistema de banco de dados é um conjunto de dados que descreve as atividades de uma organização.

Para que os dados criados pela empresa possam ser gerenciados é necessário empregar um Sistema de Gerenciamento de Banco de Dados (SGBD). Para Rodriguez, e Ferrante (2000), um SGBD é uma ferramenta que possibilita o armazenamento, manutenção, manipulação e apresentação dos dados, mantendo sua integridade e certificando sua segurança, além de que é utilizado para organizar os dados, proporcionando a realização de consultas e a atualizações das informações.

Remoaldo (2008), define o MySQL sendo um SGBD que o utiliza o modelo relacional, ou seja, armazena em sua estrutura os dados em tabelas que são compostas por colunas e linhas.

Para que os dados possam ser manipulados pelo MySQL é necessário se utilizar a linguagem de manipulação de dados como o SQL (*Structured Query Language*). Welling e Thomson (2005) enfatizam que a linguagem SQL é utilizada para armazenar e recuperar dados a partir de um banco de dados relacional, sendo utilizado pelo SGBD MySQL.

Visando facilitar no desenvolvimento de sistemas orientados a objetos, foi empregado padrões da UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), que no contexto de Neto (2009), é uma linguagem que possui notação universal para a documentação de aplicativos sobre o paradigma de orientação a objetos. Suas principais características são: facilitar o desenvolvimento e a documentação de sistemas, auxiliar na modelagem de qualquer sistema e ainda é dividido em diagramas gráficos.

Martins (2007) completa que a diagramação da UML é dividida em nove técnicas para a modelagem do sistema específico, mas apenas três são empregadas para a realização do software apresentado até o momento, são elas: diagramas de casos de uso, diagramas de sequência e diagrama de classes. A tabela 2 demonstra as informações sobre os diagramas elaborados durante o desenvolvimento do projeto.

Tabela 2: Diagramas da modelagem UML

Diagrama	O que o Diagrama Exibe	O que o Diagrama Costuma Fazer
Casos de uso	A interação entre os usuários (atores) externos e o software desenvolvido (produto)	Determinar os requisitos da empresa para o sistema
Classes	Demonstrar as condições da natureza estática de um sistema em nível de classe	Constatar os relacionamentos entre classes modeladas no sistema
Sequências	Ilustrar como os objetos interagem conforme são trocadas mensagens entre eles	Modelar o comportamento dos objetos

Fonte: Schach (2010)

Os diagramas de casos de uso representam os requisitos iniciais propostos pelo software. De acordo com Melo (2010), os diagramas de casos de uso têm por finalidade descrever uma sequência de atividades que demonstram as aplicações e comportamento de um software, através de interações com atores (usuário que interage com o sistema).

Essa diagramação permite ao desenvolvedor que levante os requisitos iniciais com o usuário, auxiliando na documentação das informações importantes para o desenvolvimento quanto produto final.

O diagrama de classe é o fundamento para o desenvolvimento de uma modelagem sobre o paradigma orientado a objetos, Tendo a finalidade descrever os atributos e métodos que compõem as classes no sistema. No entendimento de Martins (2007), o diagrama de classes tem a finalidade de apresentar a estrutura estática do software, sendo compostas pelas classes de negócio, classes de interface com o usuário e as classes de controle.

O diagrama de sequencia demonstra as interações entre os objetos que representam o sistema. Segundo Martins (2007), este diagrama determina a troca de mensagens entre as classes na realização de um caso de uso. Sendo utilizado para exibir como um cenário específico de um caso de uso será implementado no decorrer do tempo.

É relevante durante a fase de desenvolvimento de um software considerar como o usuário fará a interação com o sistema proposto, sendo aplicados conceitos da usabilidade, que auxilia a melhorar e aperfeiçoar as técnicas entre o ser humano e a máquina.

De acordo com Carrion (2008), a usabilidade é aplicada para que os usuários naveguem sem que haja dificuldades em realizar determinada tarefa por meio de uma interface gráfica. A tabela 3 apresenta as informações sobre algumas metas da usabilidade propostas pelo presente estudo.

Tabela 3: Metas desenvolvidas para a usabilidade

Metas da Usabilidade	Descrição das metas
Eficácia	Este elemento se refere como o usuário espera que o sistema desempenhe seu papel
Eficiência	Este elemento refere-se à maneira como o sistema auxilia os usuários na realização das tarefas
Segurança	Este elemento tem a finalidade oferecer proteção de possíveis condições e situações indesejáveis
Feedback	Este elemento visa retornar informações a respeito da ação que foi realizada auxiliando ao usuário continuar a executar as tarefas sem erros
Capacidade de aprendizado	Este elemento refere-se a facilidade de utilizar determinada aplicação sem que o usuário perca tempo para aprende-la
Capacidade de memorização	Este elemento refere-se em demonstrar a facilidade de lembrar como utilizar o sistema sem perder tempo

Fonte: Preece, Rogers e Sharp (2005)

Dentro do contexto de IHC (Interação Humano - Computador), existe uma preocupação com a confecção de interfaces gráficas que irão compor o sistema, sendo

aplicado alguns criterios ergonômicos que são propostos por Scapin & Bastien (1993). De acordo com Andrade (2007), os critérios ergonômicos propostos por Scapin & Bastien (1993) são divididos em 8 criterios que possuem uma subdivisão de 18 subcritérios que determinam a ergonomia de interfaces.

Segundo Galitz (1988) uma interface estando bem definida aumenta na velocidade tanto do processamento humano, mas quanto do processamento do computador, também oferecendo conforto e facilidade de uso do software. Afirma também, que por estar bem projetada possui um impacto direto na relação de uma organização com usuários e na lucratividade. Caso contrário se uma interface não estiver bem projetada diminuirá na velocidade de processamento do software, e causará alguns transtornos como maior ocorrência de erros na utilização, excesso de tempo para realizar ações no sistema.

A tabela 4 apresenta os critérios ergonômicos propostos por Scapin & Bastien (1993).

Tabela 4: Conjuntos de critérios e subcritérios propostos por Scapin & Bastien (1993)

Critérios	Subcritérios
Condução: Qualidade da ajuda e da documentação; Adequação ao aprendizado.	Convite Agrupamento e distinção por localização Agrupamento e distinção por formato Feedback imediato Legibilidade
Carga de Trabalho: redução da carga cognitiva e ações físicas do usuário.	Brevidade das entradas individuais Concisão das apresentações individuais Ações mínimas Densidade informacional
Controle explícito: Permissão que o usuário tenha para controle sobre ações do sistema que podem implicar em perda de tempo e de dados.	Ações explícitas Controle do usuário
Adaptabilidade: Propor maneiras diferentes de realizar a mesma tarefa.	Flexibilidade Consideração da experiência do usuário
Gestão de erros: Tolerância aos erros.	Proteção contra os erros Qualidade das mensagens de erros Correção dos erros
Homogeneidade/Consistência: Manter a consistência para os elementos da interface.	Homogeneidade interna a uma aplicação Homogeneidade externa a plataforma
Significados de códigos e denominações: usar códigos e denominações que deixem clara a relação de interface com as informações	

Compatibilidade: Compatível com a maneira como ele realiza a tarefa	
---------------------------------------------------------------------	--

Fonte: Cybis, Betiol e Faust (2010)

O ciclo de vida escolhido para o desenvolvimento do sistema para controle de estoque da tapeçaria foi o modelo de espiral. De acordo com Audy e Prikladnicki (2008), o ciclo de vida espiral pode ser conhecido também como paradigma de Boehm (1991), este ciclo foi desenvolvido para englobar as melhores características dos ciclos de vida do modelo clássico e incremental, ao mesmo tempo em que adiciona o elemento de análise de risco, que no caso não existe nos modelos anteriores. Este ciclo de vida define quatro atividades que foram propostas por Peters (2001), como: planejamento (requisitos e objetivos do software), análise de riscos (identificação/resolução dos riscos), engenharia (desenvolvimento do sistema) e avaliação do cliente (avaliação dos resultados propostos pela engenharia).

Ainda de acordo com Audy e Prikladnicki (2008), este ciclo é baseado nas decisões de prosseguir ou não prosseguir de acordo com a avaliação por parte do cliente com o desenvolvimento do software quanto produto final.

Etapas do Desenvolvimento do Projeto

Primeiramente há a necessidade da escolha de um ciclo de vida a ser seguido durante o desenvolvimento do software, o qual melhor se adapte ao projeto proposto. Para o desenvolvimento do sistema para controle de estoque da tapeçaria, o ciclo empregado foi o espiral.

Após a escolha de do ciclo de vida espiral, foi dado andamento no projeto e executado a primeira fase do ciclo, o qual é planejar como o projeto irá se suceder, contudo foram elencados todos os requisitos junto ao cliente e identificado as necessidades do mesmo.

A coleta de requisitos foi obtida através de métodos como entrevista aberta e observação in loco, sendo verificado junto ao cliente como era a metodologia de trabalho diária. Nesta etapa foram elaborados os diagramas de casos de uso, recomendados pela UML, com a finalidade de facilitar no levantamento de requisitos entre o desenvolvedor e a comunicação com o cliente.

A fase seguinte do ciclo de vida é a etapa de análise de riscos. Nessa fase se analisou juntamente com o cliente se as suas necessidades estavam de acordo com a coleta de requisitos para a confecção do software. Para a análise de riscos foi efetuada uma reunião juntamente com o cliente para identificar se realmente os requisitos levantados eram os necessários para o sistema, na reunião foram apontados melhorias para o sistema, ou seja, implementadas novas funcionalidades ao sistema. A ata elaborada durante a reunião e os requisitos iniciais encontra-se nos apêndices “A, B”.

A terceira etapa do ciclo de vida é a fase da engenharia. Nesta etapa foram analisadas as tecnologias que seriam empregadas para a elaboração do software. Sendo definido a linguagem de programação que iria ser aplicada, a plataforma IDE que iria auxiliar durante o processo de desenvolvimento, e o SGBD. Ainda nesta fase foram elaborados os diagramas de sequencia e de classe propostos pela UML e a modelagem logica do banco de dados.

Para a modelagem lógica do sistema foi utilizado o software MySQL Workbench 6.0. Esta ferramenta proporciona uma visualização em um ambiente gráfico, demonstrando como é a estrutura do banco de dados relacional. A modelagem do banco de dados pode ser encontrada no apêndice “T”.

A elaboração dos diagramas propostos pela UML foram desenvolvidas através da ferramenta Astah Professional. Os diagramas da UML encontram-se nos apêndices “C”, “D”, “E”, “F”, “G” “H”.

Para o desenvolvimento do sistema foi utilizado uma versão do framework desenvolvido no período da graduação com algumas modificações, sendo que para as modificações e o desenvolvimento do sistema foi utilizado o software gratuito NetBeans IDE 7.3.1

A fim de ter acesso ao esquema gráfico do banco de dados MySQL foi necessário utilizar um aplicativo web chamado phpMyAdmin. Contudo, é de suma importância executar um servidor web como o Apache, que é de código aberto, e disponibiliza apoio aos demais softwares utilizados durante o desenvolvimento do sistema. Por fim, nessa etapa, contou-se com o XAMPP, que é o pacote de aplicativos livres que agrega o Apache e interpretadores para o PHP.

Após o procedimento da escolha das tecnologias empregadas, o processo de desenvolvimento do software obteve início. Buscando uma organização dos códigos foi necessário aplicar o padrão de projeto MVC, já definido anteriormente.

Nesta fase foi levado em consideração como o usuário iria fazer a interação com o sistema proposto, obedeceram-se alguns criterios ergonômicos dispostos pela IHC, caracterizados por Scapin & Bastien (1993). Critérios ergonômicos como gestão de erros, agrupamento e distinção por localização, legibilidade e ações mínimas. Os criterios ergonômicos citados se encontram no apêndice “J”.

Durante o processo de confecção da interface gráfica, se fez uso da biblioteca Bootstrap. Esta biblioteca possui o código fonte aberto, proporcionando auxiliar na criação de *templates* e *layouts* com um designer visual adequado, além de possuir na sua composição uma biblioteca de CSS pronta, visando seguir os critérios ergonômicos propostos.

Utilizado também a linguagem JavaScript, o qual pode exibir a mesma pagina de maneiras diferentes, obtendo assim uma página mais dinâmica utilizando-se da biblioteca gratuita *Jquery*.

Nesta etapa o software foi desenvolvido como produto final, pois será avaliado pelo cliente na próxima fase do ciclo de vida.

A ultima fase do ciclo de vida espiral é a fase de avaliação do cliente. Nessa fase foi elaborado um questionário que demonstra questões sobre a implementação do software, ou seja, se o desenvolvimento obteve êxito e sobre a interface gráfica proposta. O questionário avaliativo se encontra no apêndice “K”.

Resultados

Com o desenvolvimento do software, foi possível perceber a real importância de se planejar um sistema antes de implementá-lo e se seguir um ciclo de vida, sendo que uma das etapas principais do ciclo é o levantamento de requisitos.

Durante o desenvolvimento foi possível utilizar as tecnologias que foram estudadas durante a graduação sendo que houve um maior aprofundamento nessas tecnologias, pois juntamente com a pesquisa bibliográfica foi possível identificar ferramentas e diferentes formas para a aplicação.

Com a utilização do ciclo de vida espiral, foi possível perceber a importância da análise de riscos antes da execução do início do desenvolvimento, pois se houver perda de tempo para uma análise ou remanejamento no sistema o custo ficará muito mais alto.

Após o término do desenvolvimento o cliente teve seu primeiro contato direto com o software o qual utilizou por algumas horas para poder ver se estava conforme o solicitado. Contudo, podemos ver a real importância de um levantamento de requisitos bem complexo e um planejamento sobre os riscos do desenvolvimento.

Considerações Finais

O real objetivo do estudo foi desenvolver uma aplicação baseada nas tecnologias estudadas durante a graduação, a qual foi utilizada uma específica, a linguagem PHP orientada a objetos juntamente com o banco de dados MySQL.

O software em questão foi desenvolvido para à gerencia de controle de estoque e financeira de uma empresa a qual atua-se no ramo de tapeçaria automotiva e residencial.

O sistema será implantado futuramente e será testado em prática, caso necessário futuramente serão efetuadas melhorias, sempre buscando facilitar ao cliente controle e informações.

Para a empresa a utilização do software será de suma importância, pois atualmente se leva horas para conseguir saber o ganho real da empresa, com o software ela terá essas e outras informações em segundos, até mesmo o saldo dos produtos.

Referências Bibliográficas

- ANDRADE, Antonio Luis Lordelo(2007).” Usabilidade de interfaces Web: avaliação Heurística e jornalismo on-line”. Editora: E-papers. Rio de Janeiro - RJ.
- AUDY, Jorge; PRIKLADNICKI, Rafael (2008). “Desenvolvimento distribuído de software”. Editora: Elsevier. Rio de Janeiro - RJ.
- BASTIEN; SCAPIN (1993). “Critérios Ergonômicos para Avaliação de Interfaces Homem-Computador”. Disponível em: <
http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/LabiUtil2003Crit/110prestez_A.html > Acesso em: 20 de agosto de 2013.
- CARRION, Wellington (2008). “Design para Webdesigners”. Editora: Brasport. Rio de Janeiro – RJ.
- CYBIS. Walter; Betiol, Adriana Holtz; Faust, Richard (2010). “Ergonomia e Usabilidade: conhecimentos, métodos e aplicações”. 2ª edição. Editora Novatec. São Paulo – SP.
- DALL’OGLIO, Pablo (2007). “PHP: programando com orientação a objetos”. Editora: Novatec. São Paulo – SP.
- DEITEL, H. M; DEITEL, P.J.; NIETO, T (2001). “XML – Como Programar”. Editora: Artmed. Porto Alegre – RS.

- GALITZ, Galitz.(1988) Técnicas de Formatação de Tela. Tradução de Daniel Vieira. Rio de Janeiro - RJ
- JOBSTRAIBIZER, Flávia (2009). “Criação de sites com CSS”. Editora Digerati Books São Paulo – SP.
- Laudon & Laudon, Kenneth C; Jane P (2007). “Sistemas de Informação Gerenciais”. 7ª edição. Tradução Thelma Guimarães; Revisão técnica Belmiro N. João. Editora Pearson Prentice Hall, São Paulo – SP.
- LOTTE, Roseli Imming. O Papel da informação na Organização: 2004. Disponível em: <<http://e-revista.unioeste.br/index.php/expectativa/article/download/748/633>>
- Acesso em 16 de outubro de 2013.
- MARTINS, Jose C. Cordeiro (2007). “Técnicas para gerenciamento de projetos de software”. Editora: Brasport. Rio de Janeiro – RJ.
- MELO (2010), Ana Cristina. “Desenvolvendo aplicações com UML 2.2: do conceitual à implementação”. 3ª edição. Editora Brasport, Rio de Janeiro – RJ.
- MIYAGUSKU (2007), Renata H. M. “Crie sites arrasadores”. Editora Digerati Books. São Paulo – SP.
- NETO, Oziel Moreira (2009). “Entendendo e Dominando o Java”. 3ª edição. Editora: Digerati Books. São Paulo – SP.
- NIEDERAUER, Juliano (2004). “Desenvolvendo Websites com PHP”. Editora: Novatec. São Paulo - SP.
- PETERS, JAMES F (2001).” Engenharia de Software: Teoria e Prática”. Editora: Campus. Rio de Janeiro - RJ.
- PREECE, Jenny; Rogers, Yvonne; Sharp, Helen (2005). “Design de interação: além da interação home-computador”. Tradução: Viviane Possamai. Porto Alegre – RS.
- PRESSMAN, Roger S (2006). “Engenharia de Software”. 6ª edição. Editora: Artmed. Porto Alegre – RS.
- RAMAKRISHNAN, R.; GEHRKE, J (2008). “Sistemas de Gerenciamentos de Bancos de Dados”. 3ª edição. Editora:McGraw Hill Brasil.
- REMOALDO, Pedro (2008). “O guia prático do Dreamweaver CS3 com PHP, Javascript e Ajax”. Editora: Centro Atlântico. Lisboa – PT.
- REZENDE, Denis Alcides (2002). “Tecnologia da informação: integrada a inteligência empresarial”. Editora: Atlas, São Paulo – SP.
- RODRIGUEZ, Martius V.; FERRANTE, Augustin J (2000). “Tecnologia de Informação e Gestão Empresarial”. Tradução: Washington Luiz Salles e Louise Anne N. Bonitz. Editora: E- Papers. Rio de Janeiro – RJ.
- SAMPAIO, Cleuton Melo J (2007). “Web 2.0 e Mashups: reinventando a internet”. Editora: Brasport. Rio de Janeiro – RJ.
- SCHACH, Stephen R (2010). “Engenharia de software: os paradigmas clássico e orientado a objetos”. 7ª edição. Editora: Guacira Simonelli. Porto Alegre - RS.

SILVA, Mauricio Samy (2010). “jQuery: a biblioteca do programador JavaScript”.
Editora Novatec, São Paulo – SP.

WELLING, L; THOMSON, L (2005). “PHP e MySQL Desenvolvimento Web”. 6ª
Edição. Editora: Elsevier. Rio de Janeiro – RJ.

Apêndices

- Requisitos
 - Apêndice A: Requisitos iniciais:
Levantamento de Requisitos

Sistema para controle de Tapeçaria

Prudentópolis, 17 de maio de 2013.

1. O sistema deve permitir cadastros de Clientes;
2. Cadastro de fornecedores;
3. Cadastro de produtos;
4. Efetuar Vendas;
5. Visualizar contas a receber;
6. Visualizar contas já recebidas
7. Opção para efetuar um pagamento



Antonio Olívio Bini

Proprietário Tapeçaria Bini



Selmo Andrei Bobato

Acadêmico Análise de Sistemas

81 983 040/0001-46 7

ANTÔNIO OLÍVIO BINI - ME

Rua Prudente de Moraes, 140
Vila Nova

Cep 84400

○ Apêndice B: Requisitos pós a análise de riscos:

ANALISE DE REQUISITOS E RISCOS DO DESENVOLVIMENTO

Durante o dia 07 de junho de 2013, reuniram as seguintes pessoas Selmo Andrei Bobato o qual é acadêmico do curso de Tecnologia em Análise e desenvolvimento de sistemas da Faculdade Guairacá e o senhor Antonio Olivio Bini o qual é proprietário da empresa Tapeçaria Bini. O objetivo desta reunião foi a decisão dos requisitos do sistema para controle de estoque da tapeçaria. Após uma análise dos requisitos levantados na primeira reunião o acadêmico apresentou melhorias e novas funcionalidades para o sistema ficar um pouco mais complexo. Atualmente o sistema encontrava-se com os seguintes requisitos levantados:

1. Cadastro de Clientes;
2. Cadastro de fornecedores;
3. Cadastro de produtos;
4. Efetuar Vendas;
5. Visualizar contas a receber;
6. Visualizar contas já recebidas;
7. Opção para efetuar um pagamento.

As melhorias propostas pelo acadêmico foram às inserções de mais alguns itens, como:

1. Controle de usuários;
2. Relatório de Vendas;
3. Relatório de Estoque;
4. Cadastro de Matriz/filial;
5. Relatório Comparativo Matriz/Filial.

Após a apresentação das melhorias houve um debate sobre a necessidade da implementação dessas funcionalidades e o risco do desenvolvimento e utilização do sistema sem as mesmas. Contudo o proprietário da empresa optou por não possuir um cadastro de matriz/filial, pois atualmente o mesmo efetua compras para um estabelecimento, o qual a contabilidade não é efetuada de forma separada.

Decidido então a continuidade do desenvolvimento com os seguintes requisitos.

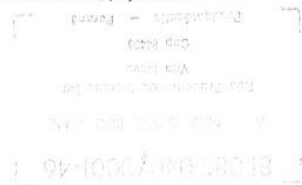
1. Cadastro de Clientes;
2. Cadastro de Fornecedores;
3. Cadastro de produtos;
4. Efetuar Vendas;
5. Visualizar contas a receber;
6. Visualizar contas já recebidas;
7. Opção para efetuar um pagamento
8. Controle de usuários;
9. Relatório de Vendas;
10. Relatório de Estoque;


Antonio Olivio Bini

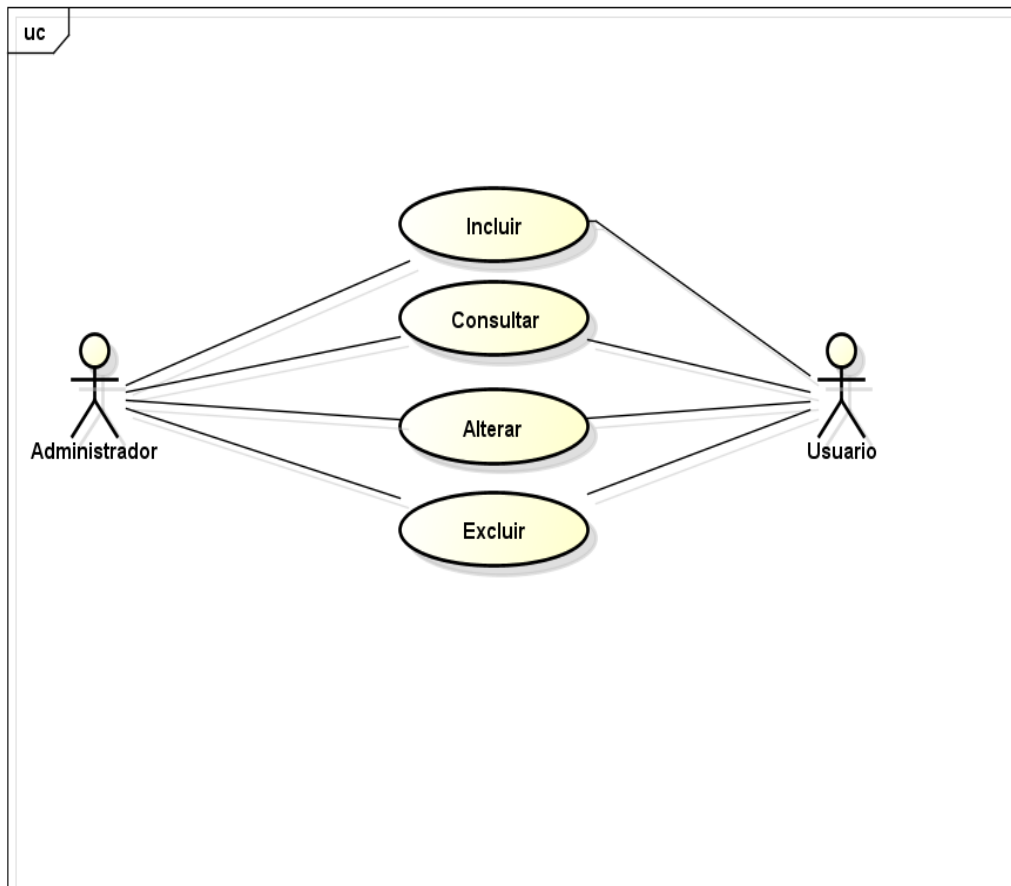
Proprietário Tapeçaria Bini


Selmo Andrei Bobato

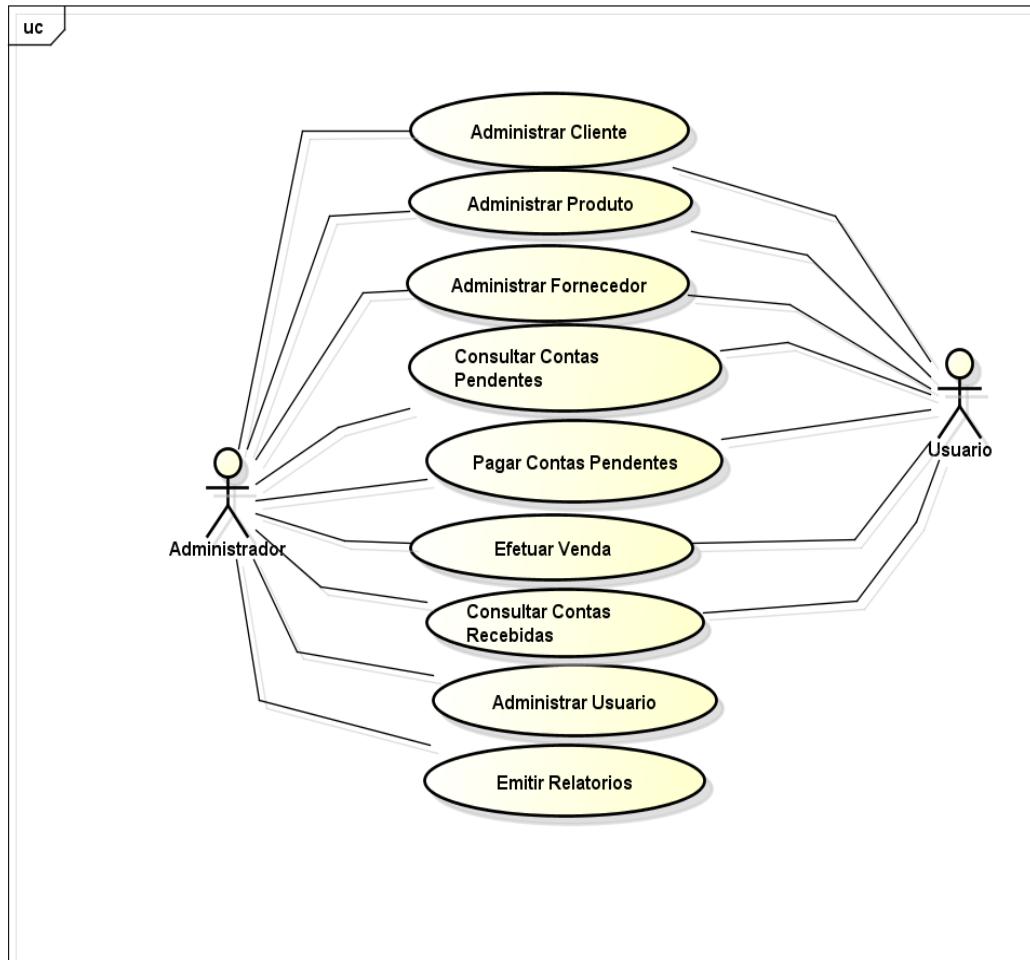
Acadêmico Análise de Sistemas



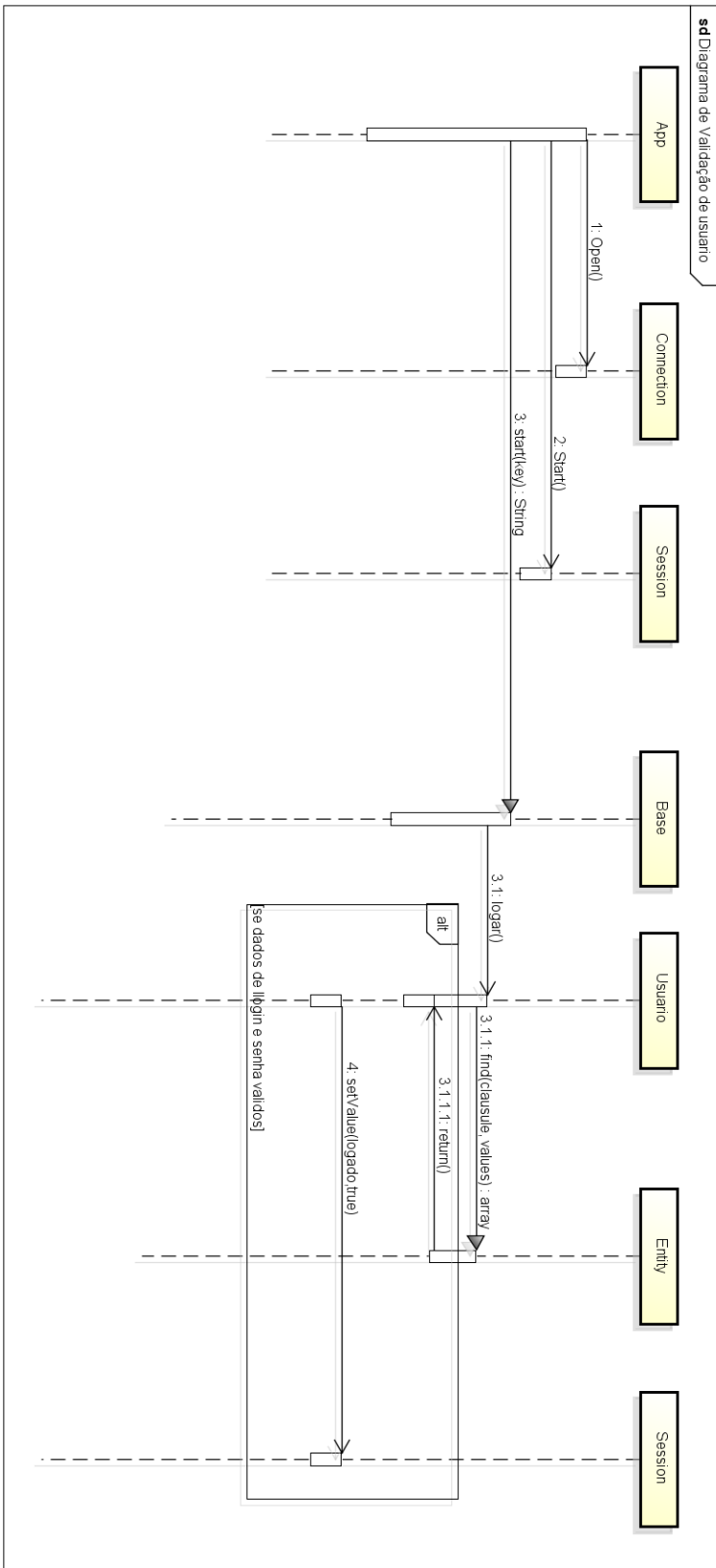
- Diagrama de Casos de Uso
 - Apêndice C: Diagrama Caso de Uso Administrar



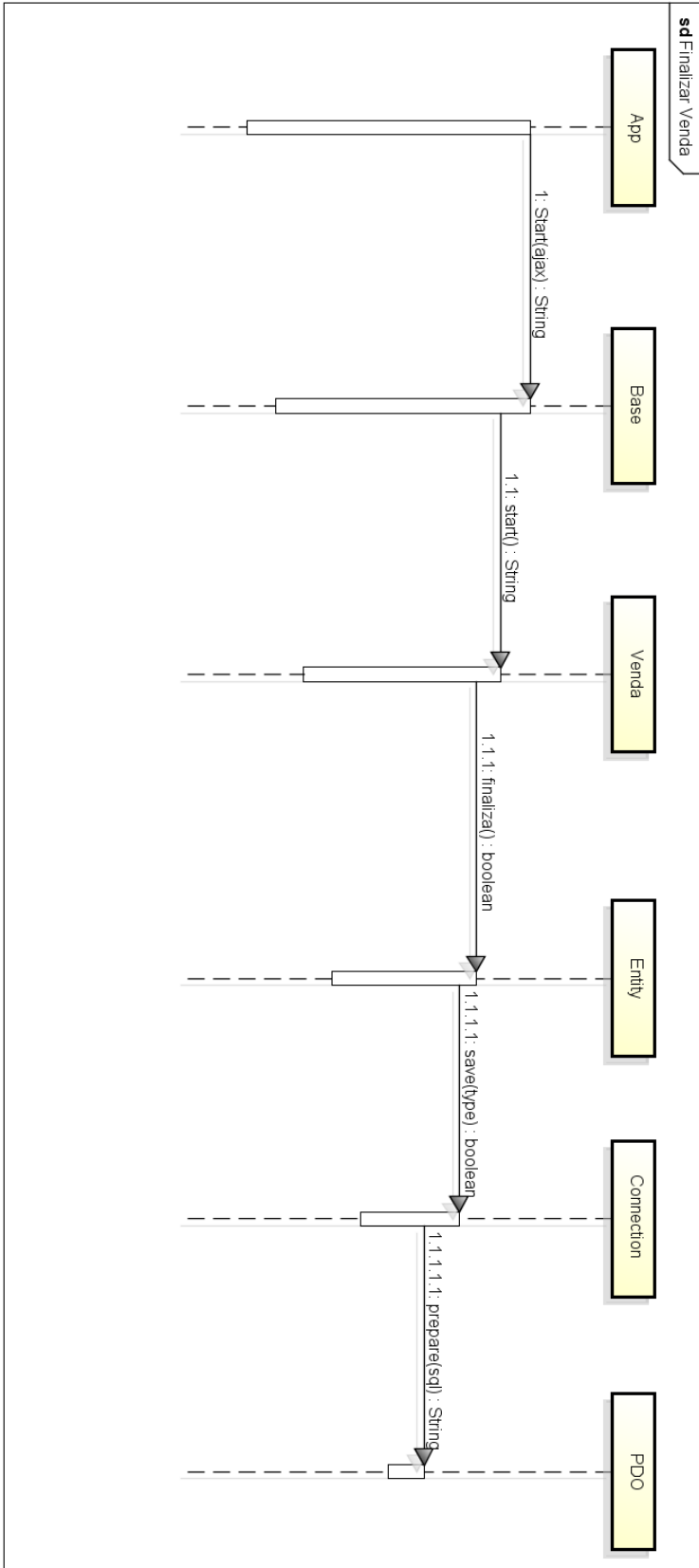
○ Apêndice D: Diagrama Caso de Uso Geral



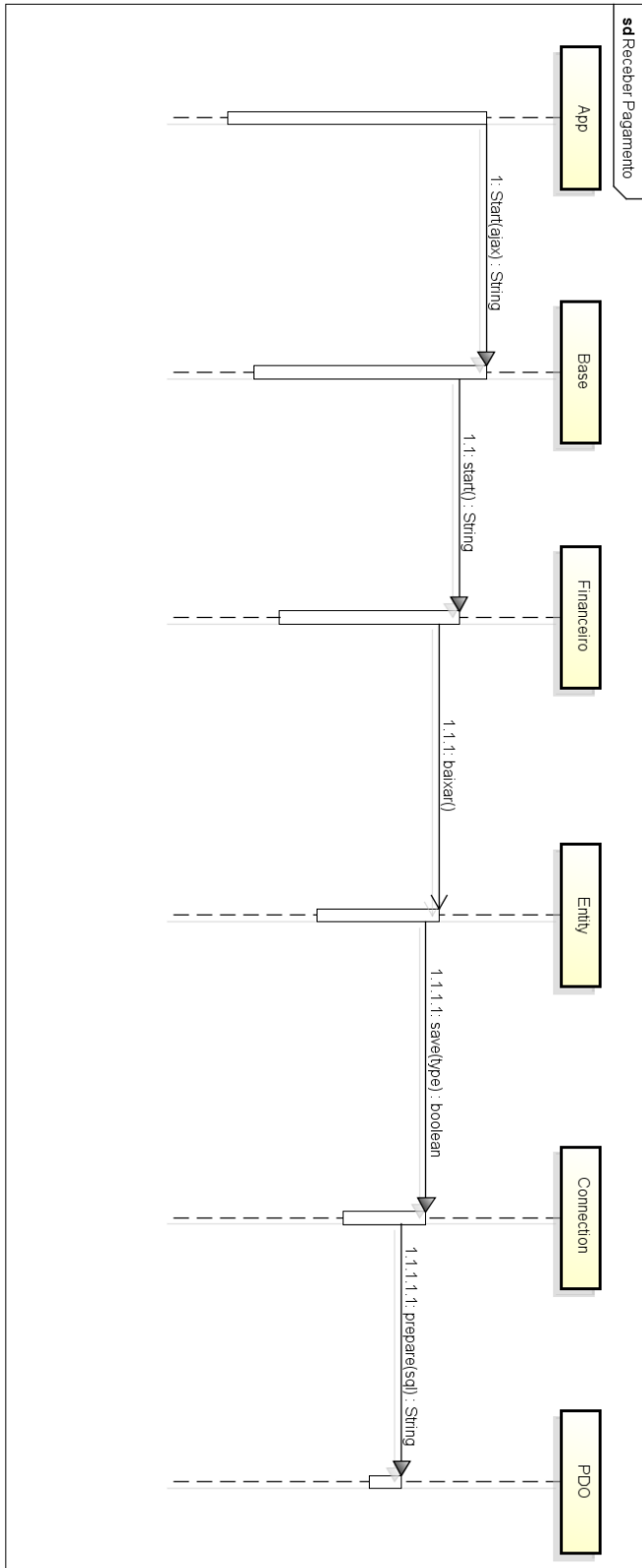
- Diagramas de Sequências
 - Apêndice E: Diagrama de Sequência Validação de Usuário



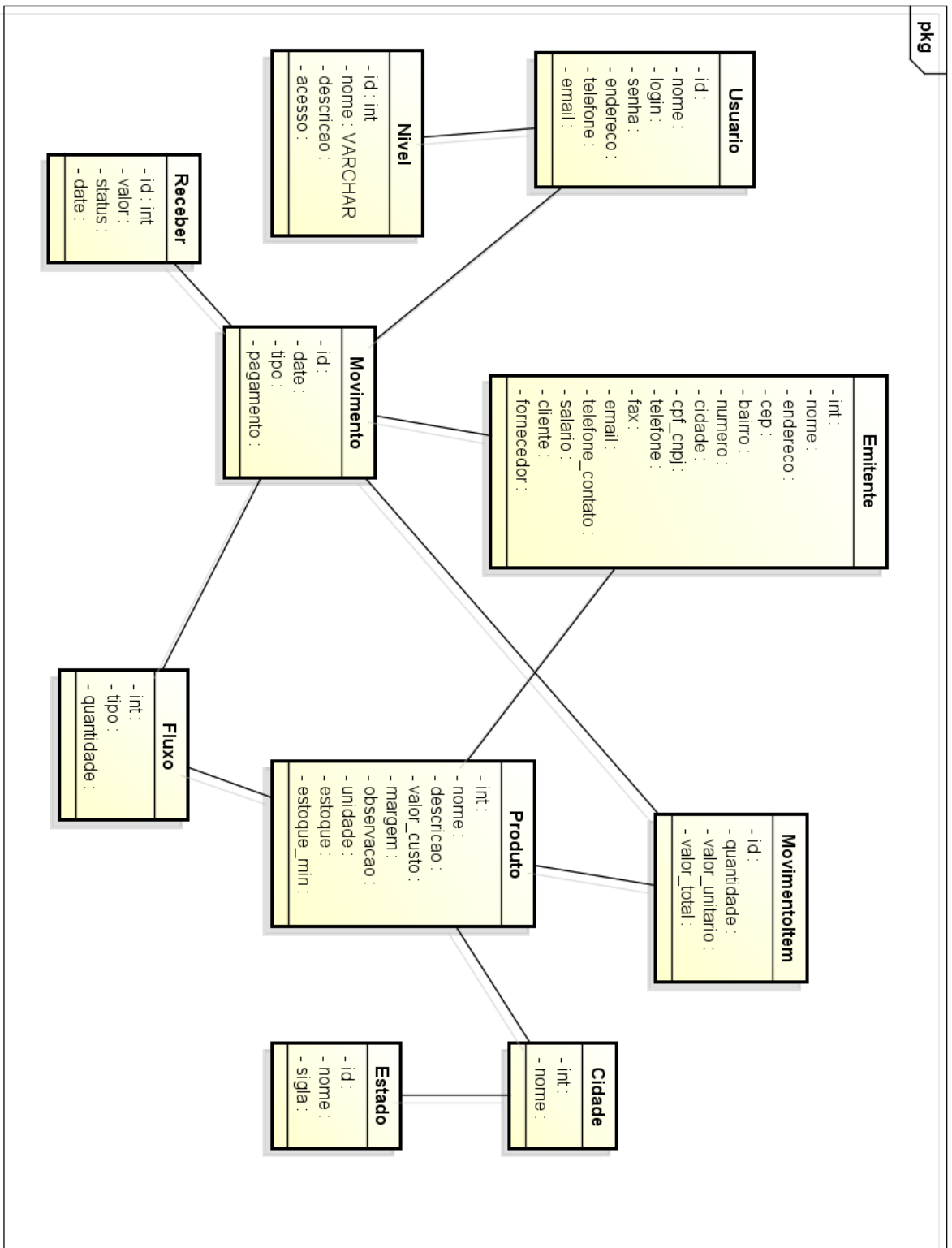
○ Apêndice F: Diagrama de Sequência Finalizar Venda



○ Apêndice G: Diagrama de Sequência Receber Pagamento

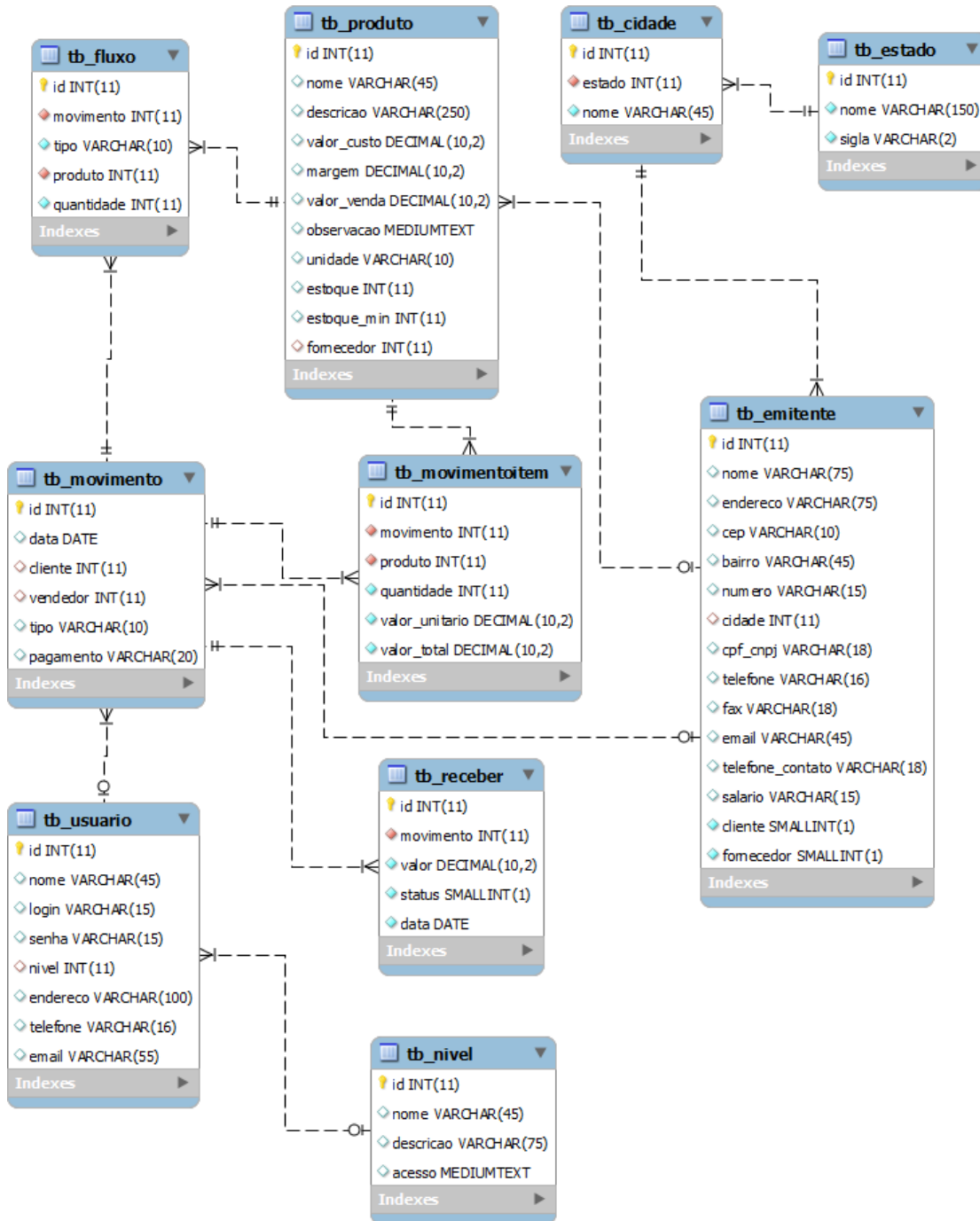


- Diagrama de Classes
 - Apêndice H: Diagrama de Classes



- Banco de Dados

- Apêndice I: Modelagem Lógica do Banco de Dados



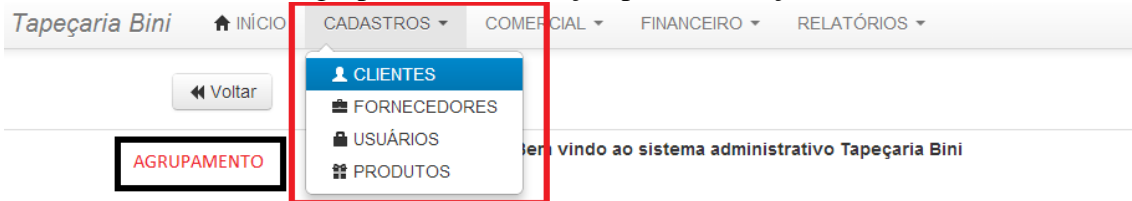
- Usabilidade

- Apêndice J: Critérios Ergonômicos

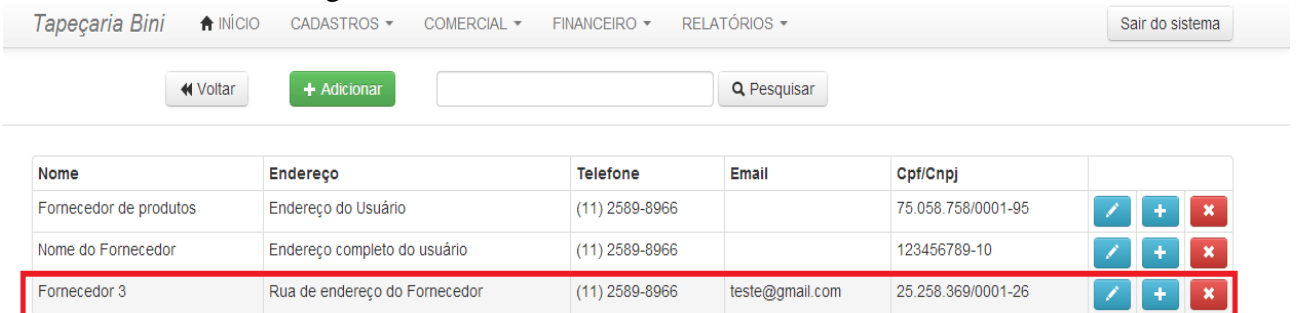
- Gestão de Erros



- Agrupamento e distinção por localização




- Legibilidade



Legibilidade - Linhas escuras da tabela com diferenças de cor

- Ações mínimas

RELATÓRIO DE ESTOQUE

Por fornecedor	Fornecedor de produtos 
Por produto	Selecione o fornecedor...
	Fornecedor de produtos
	Nome do Fornecedor
	Fornecedor 3
	Anselmo Bobato
	Kelly Janaina Bobato
	Edimar Bobato

- Questionário Avaliativo
 - Apêndice K: Questionário avaliativo do cliente

Questionário de Avaliação do Sistema

O questionário abaixo tem por finalidade de o usuário avaliar o sistema em que foi desenvolvido. As questões abaixo têm por finalidade avaliar o aprendizado, satisfação e eficiência do cliente juntamente ao sistema.

Capacidade	Muito	Pouco	Não
A aplicação empregada é de fácil entendimento?	<input checked="" type="checkbox"/>		

Aprendizado	Muito	Pouco	Não
A utilização do sistema é de fácil aprendizado?	<input checked="" type="checkbox"/>		
É fácil a memorização das tarefas?	<input checked="" type="checkbox"/>		

Interface	Muito	Pouco	Não
É agradável a visualização da interface?	<input checked="" type="checkbox"/>		
Os formulários de cadastros estão bem dispostos?	<input checked="" type="checkbox"/>		
Os menus estão organizados e de forma fácil para localização?	<input checked="" type="checkbox"/>		
A navegação do sistema é fácil?	<input checked="" type="checkbox"/>		

Tempo de Resposta	Muito	Pouco	Não
O sistema responde de forma satisfatória?	<input checked="" type="checkbox"/>		

Funcionalidades	Muito	Pouco	Não
O sistema contém as funcionalidades de acordo com os requisitos?	<input checked="" type="checkbox"/>		
Correspondem as expectativas esperadas pelo usuário?	<input checked="" type="checkbox"/>		
É possível localizar falhas quando ocorrem?			<input checked="" type="checkbox"/>
É fácil compreender os resultados iniciais do sistema?	<input checked="" type="checkbox"/>		
É fácil inserir dados?	<input checked="" type="checkbox"/>		

Prudentópolis, 28 de outubro de 2013.


Antonio Olivio Bini

Proprietário Tapeçaria Bini